In-Class Activity 03-1

1 Activity 1 - Writing MIPS

It’s time to write some basic MIPS instructions. (Do this Activity after I’ve introduced MIPS to the class!)

Assume that the following variables are in the following registers:

- alpha $s0
- bravo $s1
- charlie $s2

Write a series of instructions which will perform the following calculations. You may modify any $tX registers, but do not modify any $sX registers (except the one you’re required to modify).

- $s3 = alpha + bravo + charlie
- $s4 = - bravo - charlie
- $s5 = charlie - (alpha + bravo)
- $s6 = bravo*2 + charlie*3

2 Activity 2 - Loads and Stores

Recall that we’ve seen the LA (load address) instruction before:

1a $s0, variable

Using that - and the LW,SW instructions - practice reading and writing variables in memory. Convert the following C snippet to MIPS; assume that all of the variables are words:

- foo = bar;
- baz++;  
- fred = -3;
3 Activity 3 - Flattening Branches

As you’ve seen me do in the slides, “flatten” each of these if() statements; that is, rewrite it so that it uses labels and C’s goto statement.

Snippet 1

```c
if (x == y)
    z++;  
```

Snippet 2

```c
if (one == two)
    x++;  
else if (three == four)
    y++;  
else if (five == six)
    z++;  
else
    x = y = z = 0;  
```

Snippet 3

```c
if (a < b)
{
    if (b < c)
        print("In order!\n");
    else
        print("Almost there!\n");
}  
```
Snippet 4

```c
switch (x)
{
    case 10:
        x = -1;
        break;

    case 12:
        y = 13;
        // intentional fall-through to the next case!
    case 30:
        z = 1;
        break;

    default:
        printf("ERROR!\n");
        break;
}
```

Snippet 5

```c
if (x == y)
    z++;
```

(You probably have never seen the **conditional operator** before - but it’s present in C, Java, and many other languages. Look up “Conditional Operator” on Wikipedia, or follow this link: [https://en.wikipedia.org/wiki/%3F:#C](https://en.wikipedia.org/wiki/%3F:#C))